CLAIMS

What is claimed is:

1. A method of finding byte synchronization by appending a synchronization symbol to random data.

2. The method of claim 1, further comprising opening a window at a certain point in the data and attempting to find the synchronization symbol by sliding the window until the synchronization symbol is found.

3. The method of claim 2, wherein said symbol comprises a synchronization pattern, and the pattern is considered found when the window is below a predetermined threshold in Hamming distance to the pattern.

4. The method of claim 1, wherein a pattern of a data format having said synchronization symbol and said random data, includes a bit-sync field.

5. The method of claim 1, wherein a data block format includes said random data, a first sync field, and first and second byte-sync fields, such that if an anomalous condition destroys said first byte-sync field and a portion of the random data, the

second byte-sync field retrieves byte-sync, wherein synchronizing over the random data is performed.

6. The method of claim 5, wherein the random data portion is encoded using a modulation code.

7. The method of claim 5, further comprising utilizing a byte-sync pattern such that such selected pattern maximizes a minimum distance when slided over itself.

8. The method of claim 1, wherein a 4 byte byte-sync pattern is selected, such that by opening a window 2 bytes before the pattern and closing it 2 bytes after the pattern, the minimum distance is 12, representing that up to five errors are tolerated and that six errors are detected.

9. The method of claim 1, wherein the predetermined pattern chosen is

00000001 10110111 11101010 00011000

and a window is opened 2 bytes before said pattern and closed 2 bytes after said pattern.

10. The method of claim 1, wherein said random data is interposed between first and second byte-synchronization fields.

11. The method of claim 10, wherein the first and second byte-sync fields have different patterns.

12. The method of claim 10, wherein the first and second byte-sync fields have substantially identical patterns.

13. A system for finding byte synchronization, comprising:

a synchronization symbol; and

a unit for appending said synchronization symbol to random data.

14. The system of claim 13, further comprising means for opening a window at a certain point in the data and attempting to find the synchronization symbol by sliding the window until the synchronization symbol is found.

15. The system of claim 13, wherein said symbol comprises a synchronization pattern, and the pattern is considered found when the window is below a predetermined threshold in Hamming distance to the pattern.

16. The system of claim 13, wherein a pattern of a data format having said synchronization symbol and said random data, includes a bit-sync field.

17. The system of claim 13, wherein a data block format includes said random data, a first sync field, and first and second byte-sync fields, such that if an anomalous condition destroys said first byte-sync field and a portion of the random data, the second byte-sync field retrieves byte-sync, and

wherein synchronizing over the random data is performed.

18. The system of claim 17, wherein the random data portion is encoded using a modulation code.

19. The system of claim 17, further comprising means for utilizing a byte-sync pattern such that such selected pattern maximizes a minimum distance when slided over itself.

20. The system of claim 13, wherein a 4 byte byte-sync pattern is selected, such that by opening a window 2 bytes before the pattern and closing it 2 bytes after the pattern, the minimum distance is 12 representing that up to five errors are tolerated and that six errors are detected.

21. The system of claim 13, wherein the predetermined pattern chosen is

00000001 10110111 11101010 00011000

and a window is opened 2 bytes before and 2 bytes after the correct pattern is located.

22. The system of claim 13, wherein said random data is interposed between first and second byte-synchronization fields.

23. The system of claim 22, wherein the first and second byte-sync fields have different patterns.

24. The system of claim 22, wherein the first and second byte-sync fields have substantially identical patterns.

25. An encoder for finding byte synchronization, comprising:

a synchronization symbol; and

a unit for appending said synchronization symbol to random data.

26. A format for a contiguous data block of a disk drive, a controllably positionable head for storing user data in the data block, and a read channel

responsive to the head for retrieving the user data from the data block, the data block format comprising:

a bit-sync field having a first sync pattern for synchronizing the read channel to enable the read channel to recover the user data from the data block; and

a random data field proximate said bit-sync field.

27. The format of claim 26, further comprising:

a first byte-sync field adjacent said bit-sync field; and

a second byte-sync field proximate said first byte-sync field.

28. The format of claim 27, wherein said random data field is interposed between said first and second byte-sync fields.

29. A format for a contiguous data block of a disk drive, a controllably positionable head for storing user data in the data block, and a read channel responsive to the head for retrieving the user data from the data block, the data block format comprising:

a sync field having a first sync pattern for synchronizing the read channel to enable the read channel to recover the user data from the data block;

a first byte-sync field adjacent said sync field;

a second byte-sync field proximate said first byte-sync field; and

a random data field interposed between said first and second byte-sync fields.

30. A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of finding synchronization, comprising:

appending a synchronization symbol to random data.

31. The signal-bearing medium of claim 30, further comprising:

finding a potentially noisy version of said synchronization symbol by opening a window in said random data.

32. The method of claim 1, further comprising:

detecting whether the synchronization pattern, or a noisy version thereof, is present in the random data preceding said synchronization pattern.

33. The method of claim 32, further comprising:

when said synchronization pattern or said noisy version thereof are detected in data, forming an artificial error to remove said synchronization pattern or said noisy version thereof.

34. The method of claim 2, further comprising:

detecting whether the synchronization pattern, or a noisy version thereof, is present in the random data preceding said synchronization pattern.

35. The method of claim 34, further comprising:

when said synchronization pattern or said noisy version thereof are detected, forming an artificial error to remove said synchronization pattern or said noisy version thereof.

36. The method of claim 1, further comprising:

when said synchronization pattern or said noisy version thereof are detected, registering this information and informing the sliding window that a first occurrence of such a symbol is not the synchronization pattern.

37. The method of claim 36, further comprising:

when said synchronization pattern or said noisy version thereof are detected, registering this information and informing the sliding window that a first occurrence of such a symbol is not the synchronization pattern.